

# CHAPTER 1: THE DATABASE ENVIRONMENT AND

**Modern Database Management**  
**11<sup>th</sup> Edition**

**Jeffrey A. Hoffer, V. Ramesh,  
Heikki Topi**

---

# OBJECTIVES

---

- ✖ Define terms
- ✖ Name limitations of conventional file processing
- ✖ Explain advantages of databases
- ✖ Identify costs and risks of databases
- ✖ List components of database environment
- ✖ Identify categories of database applications
- ✖ Describe database system development life cycle
- ✖ Explain prototyping and agile development approaches
- ✖ Explain roles of individuals

# DEFINITIONS

---

- ✖ Database: organized collection of logically related data
- ✖ Data: stored representations of meaningful objects and events
  - + Structured: numbers, text, dates
  - + Unstructured: images, video, documents
- ✖ Information: data processed to increase knowledge in the person using the data
- ✖ Metadata: data that describes the properties and



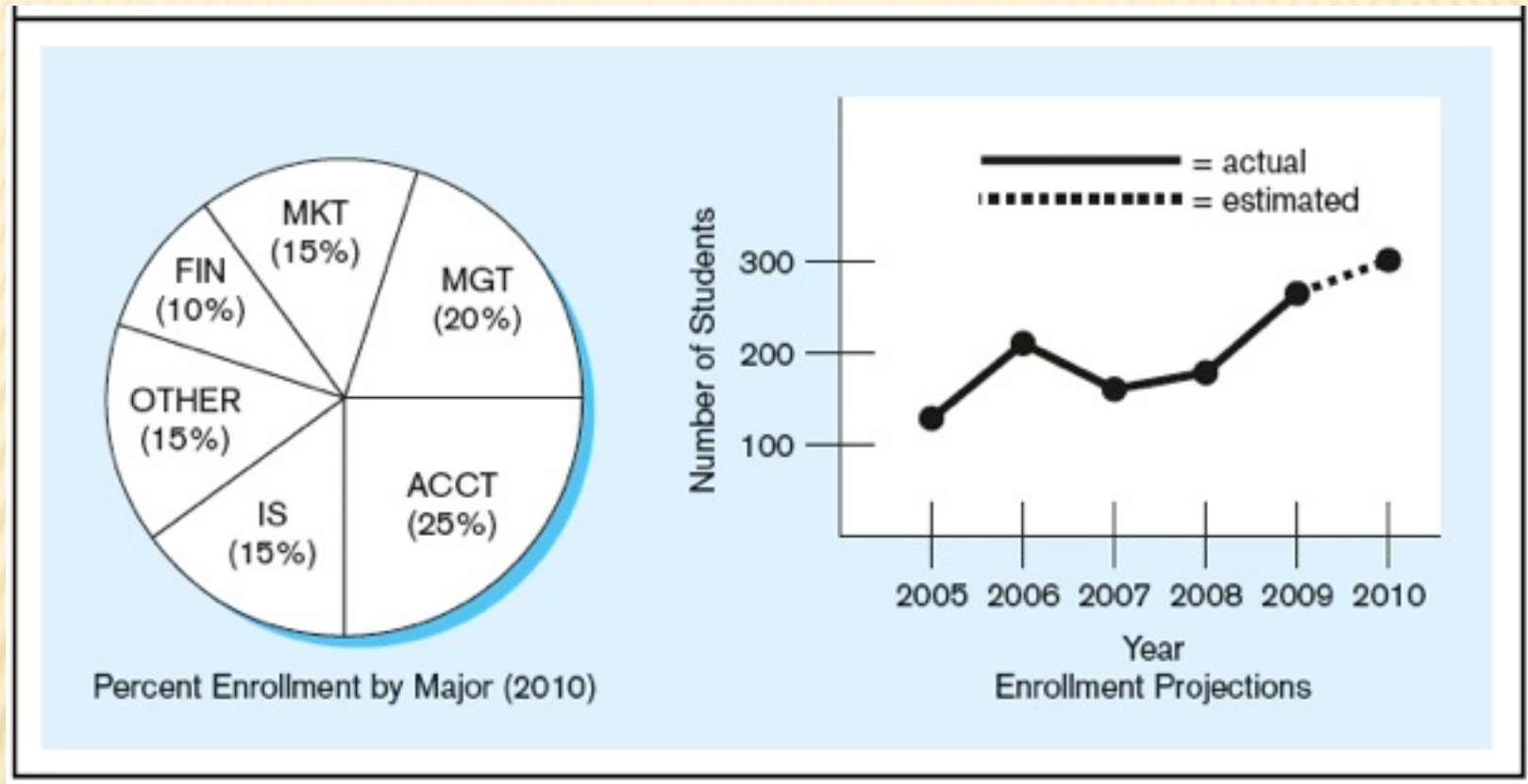
## Figure 1-1a Data in context

Class Roster			
Course:	MGT 500 Business Policy	Semester:	Spring 2010
Section:	2		
<u>Name</u>	<u>ID</u>	<u>Major</u>	<u>GPA</u>
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

**Context helps users understand data**



## Figure 1-1b Summarized data



**Graphical displays turn data into useful information that managers can use for decision making and interpretation**

**TABLE 1-1** Example Metadata for Class Roster

Data Item		Metadata				
Name	Type	Length	Min	Max	Description	Source
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

**Descriptions of the properties or characteristics of the data, including data types, field sizes, allowable values, and data context**

# DISADVANTAGES OF FILE PROCESSING

---

## ✖ **Program–Data Dependence**

- + All programs maintain metadata for each file they use

## ✖ **Duplication of Data**

- + Different systems/programs have separate copies of the same data

## ✖ **Limited Data Sharing**

- + No centralized control of data

## ✖ **Lengthy Development Times**

- + Programmers must design their own file formats

## ✖ **Excessive Program Maintenance**

- + 80% of information systems budget

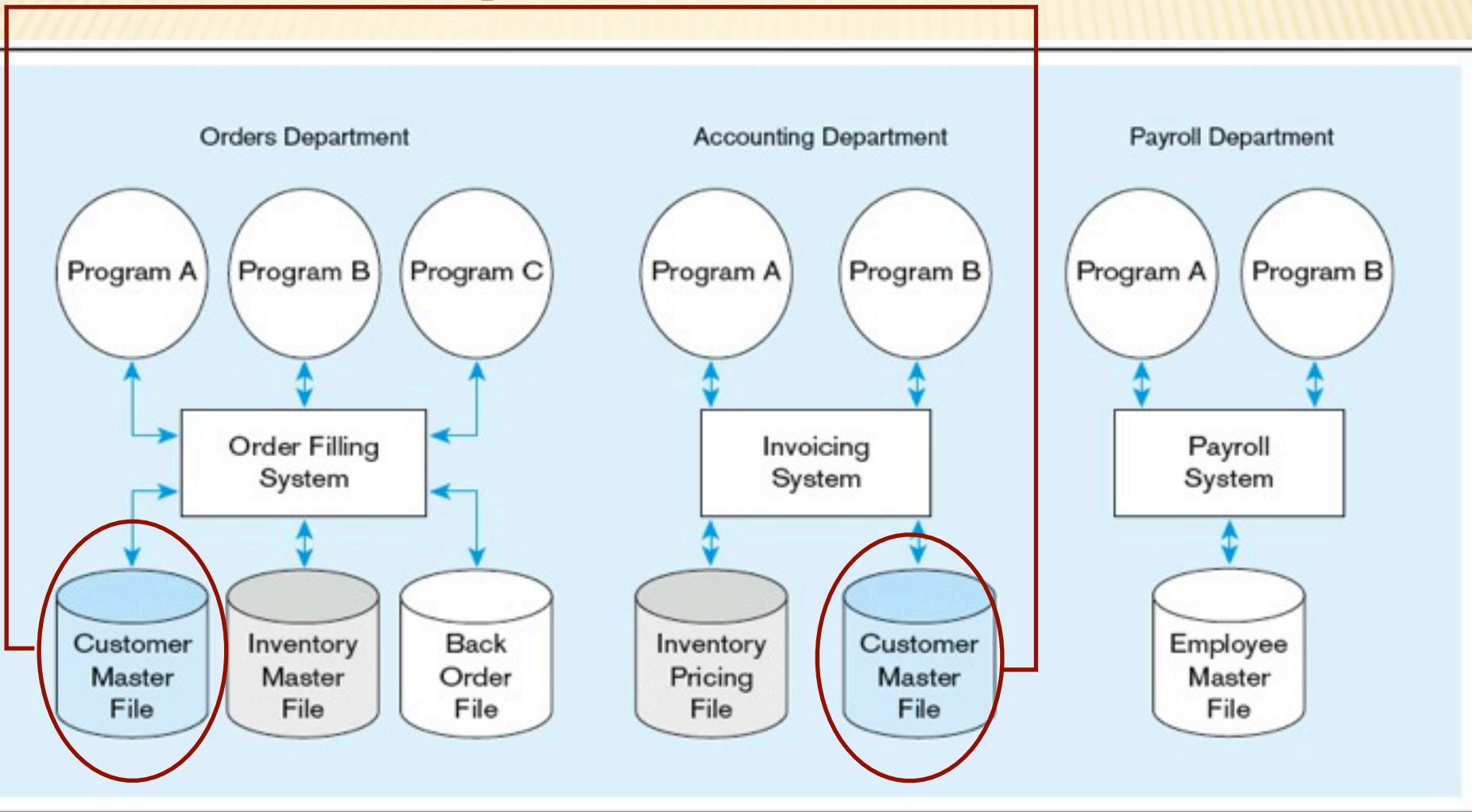


# PROBLEMS WITH DATA DEPENDENCY

---

- ✗ Each application programmer must maintain his/her own data
- ✗ Each application program needs to include code for the metadata of each file
- ✗ Each application program must have its own processing routines for reading, inserting, updating, and deleting data
- ✗ Lack of coordination and central control
- ✗ Non-standard file formats

# Duplicate Data



**FIGURE 1-2** Old file processing systems at Pine Valley Furniture Company

9

9

# PROBLEMS WITH DATA REDUNDANCY

---

- ✗ Waste of space to have duplicate data
- ✗ Causes more maintenance headaches
- ✗ The biggest problem:
  - + **Data changes in one file could cause inconsistencies**



# SOLUTION: THE DATABASE

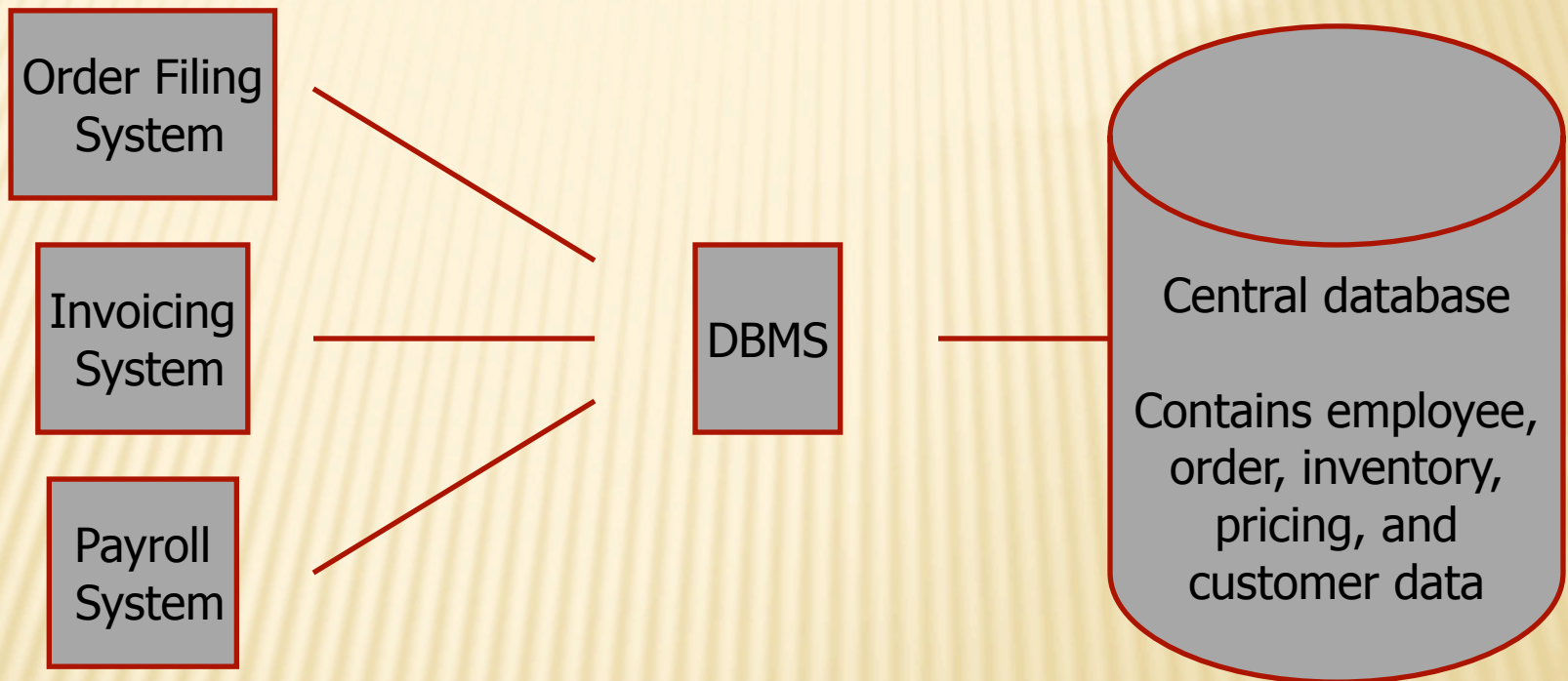
---

- ✖ Central repository of shared data
- ✖ Data is managed by a controlling agent
- ✖ Stored in a standardized,

Requires a Database Management System (DBMS)

# DATABASE MANAGEMENT SYSTEM

- A software system that is used to create, maintain, and provide controlled access to user databases



DBMS manages data resources like an operating system manages hardware resources

# ADVANTAGES OF THE DATABASE APPROACH

---

- ✗ Program–data independence
- ✗ Planned data redundancy
- ✗ Improved data consistency
- ✗ Improved data sharing
- ✗ Increased application development productivity
- ✗ Enforcement of standards
- ✗ Improved data quality
- ✗ Improved data accessibility and responsiveness
- ✗ Reduced program maintenance



# COSTS AND RISKS OF THE DATABASE APPROACH

---

- ✗ New, specialized personnel
- ✗ Installation and management cost and complexity
- ✗ Conversion costs
- ✗ Need for explicit backup and recovery
- ✗ Organizational conflict

# ELEMENTS OF THE DATABASE

---

## ✗ Data models

- + Graphical system capturing nature and relationship of data
- + Enterprise Data Model-high-level entities and relationships for the organization
- + Project Data Model-more detailed view, matching data structure in database or data warehouse

## ✗ Entities

- + Noun form describing a person, place, object, event, or concept
- + Composed of attributes

## ✗ Relationships

- + Between entities
- + Usually one-to-many (1:M) or many-to-many (M:N)

## ✗ Relational Databases

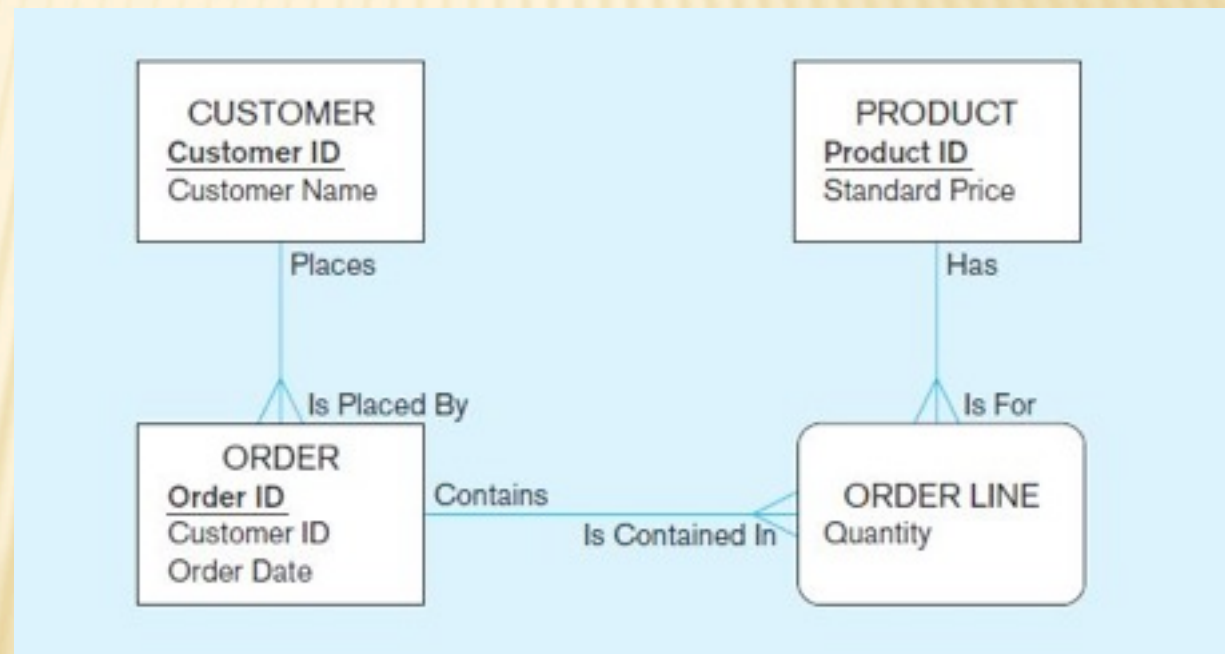
- + Database technology involving tables (relations) representing entities and primary/foreign keys representing relationships

# Figure 1-3 Comparison of enterprise and project level data models

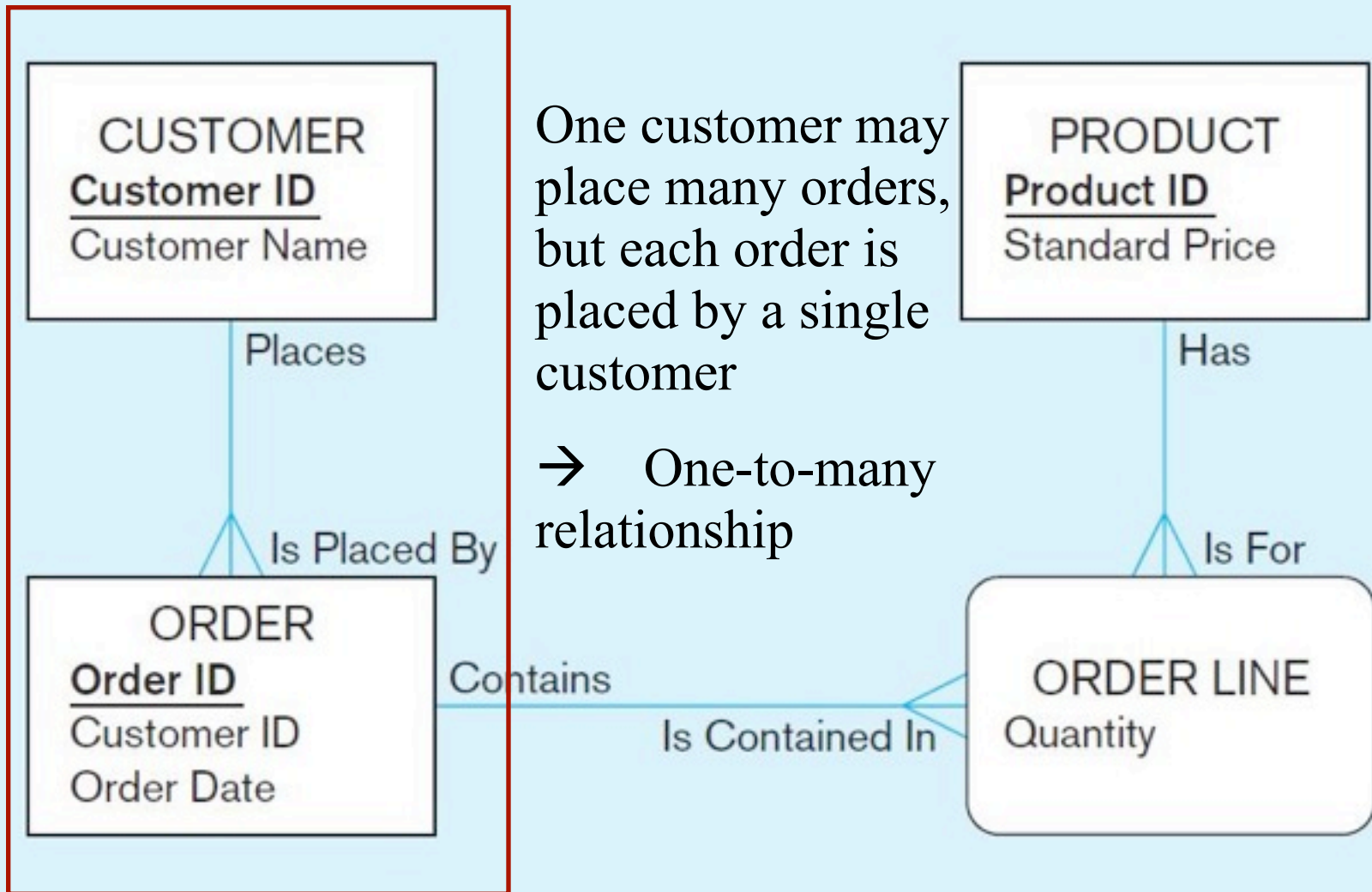
Segment of an enterprise data model

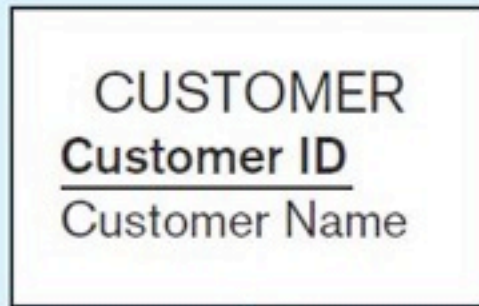


Segment of a project-level data model









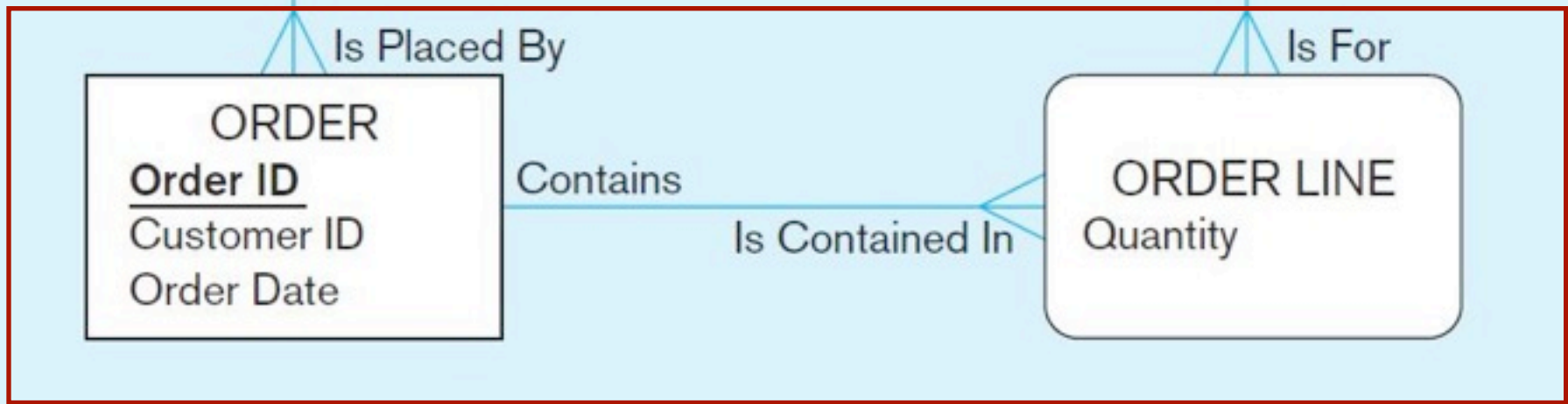
One order has many order lines; each order line is associated with a single order

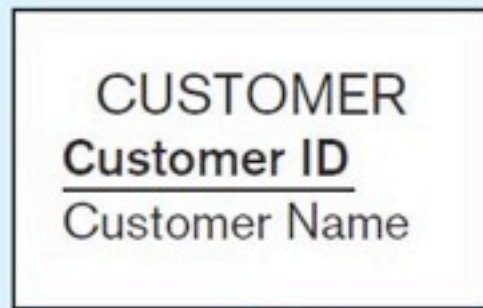


Places

Has

→ One-to-many relationship





Places

Is Placed By



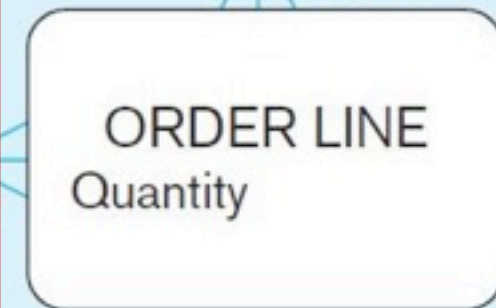
Contains

Is Contained In



Has

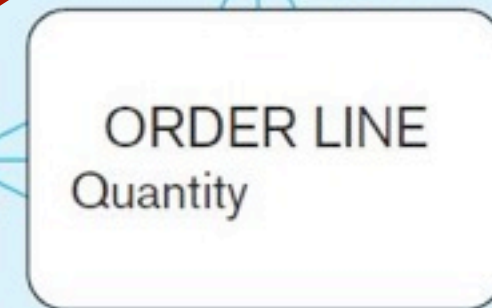
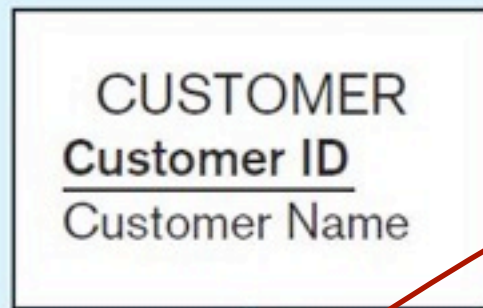
Is For



One product can be in many order lines, each order line refers to a single product

→ One-to-many relationship





Places

Has

Is Placed By

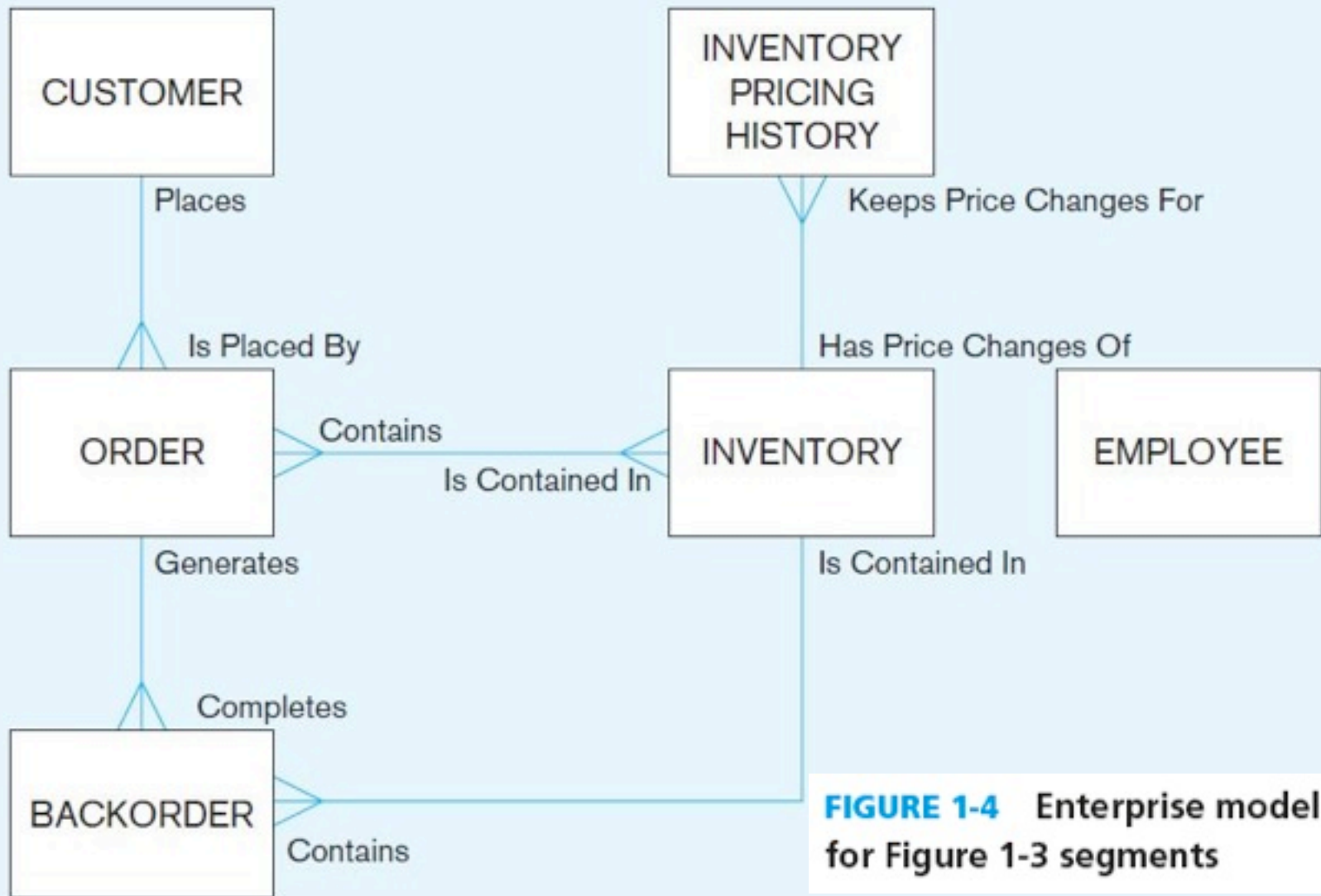
Is For

Contains

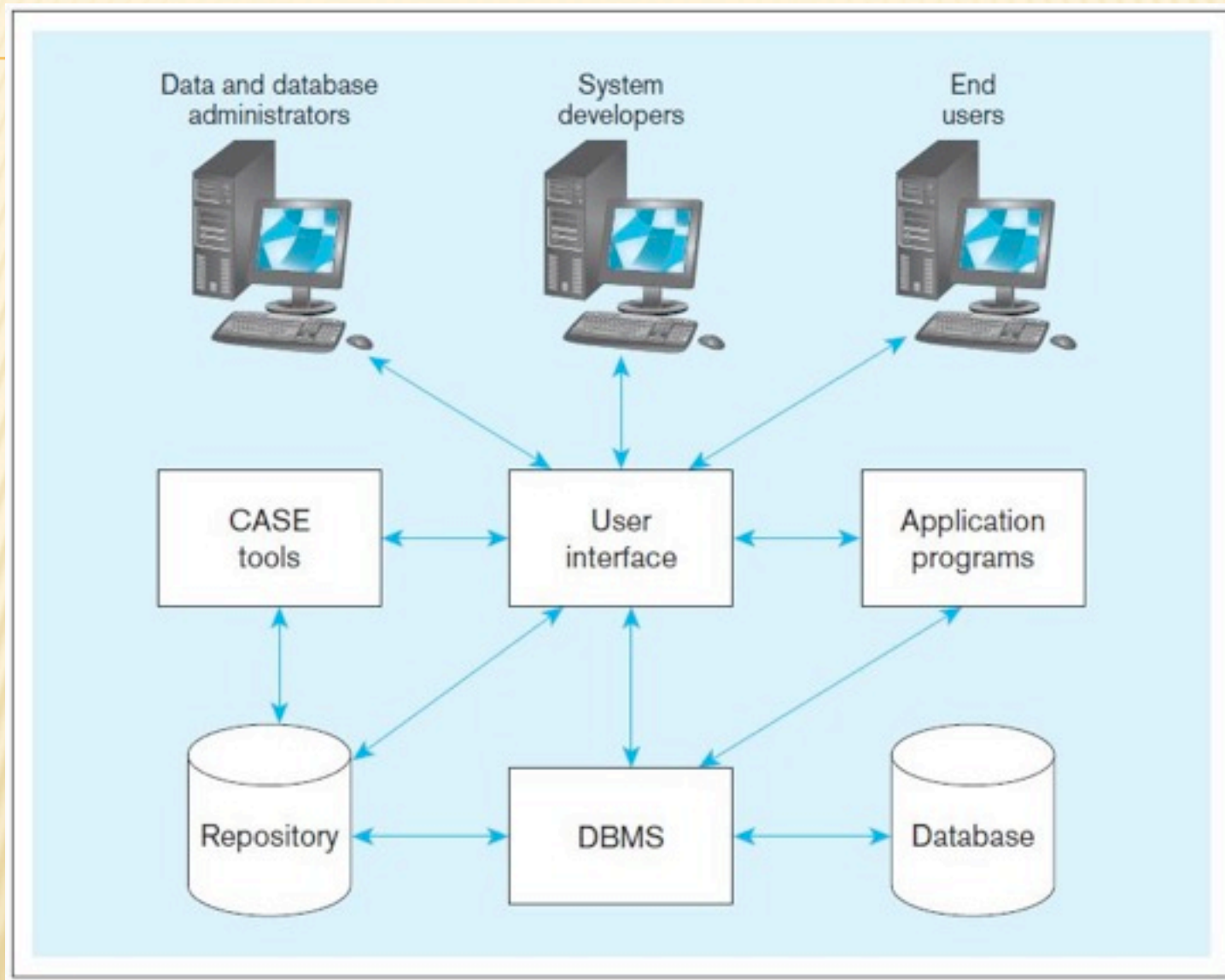
Is Contained In

Therefore, one  
order involves  
many products and  
one product is  
involved in many  
orders

⇒ Many-to-  
many relationship



# Figure 1-5 Components of the Database Environment





# COMPONENTS OF THE DATABASE ENVIRONMENT

---

- ✗ **CASE Tools**—computer-aided software engineering
- ✗ **Repository**—centralized storehouse of metadata
- ✗ **Database Management System (DBMS)** —software for managing the database
- ✗ **Database**—storehouse of the data
- ✗ **Application Programs**—software using the data
- ✗ **User Interface**—text and graphical displays to users
- ✗ **Data/Database Administrators**—personnel responsible for maintaining the database
- ✗ **System Developers**—personnel responsible for designing databases and software
- ✗ **End Users**—people who use the applications and databases

# ENTERPRISE DATA MODEL

---

- ✖ First step in the database development process
- ✖ Specifies scope and general content
- ✖ Overall picture of organizational data at high level of abstraction
- ✖ Entity–relationship diagram
- ✖ Descriptions of entity types
- ✖ Relationships between entities
- ✖ Business rules



**FIGURE 1-6** Example business function-to-data entity matrix

Business Functions \ Data Entity Types	Customer	Product	Raw Material	Order	Work Center	Work Order	Invoice	Equipment	Employee
Business Planning	X	X						X	X
Product Development		X	X		X			X	
Materials Management		X	X	X	X	X		X	
Order Fulfillment	X	X	X	X	X	X	X	X	X
Order Shipment	X	X		X	X		X		X
Sales Summarization	X	X		X			X		X
Production Operations		X	X	X	X	X		X	X
Finance and Accounting	X	X	X	X	X		X	X	X
X = data entity is used within business function									



# TWO APPROACHES TO DATABASE AND IS

---

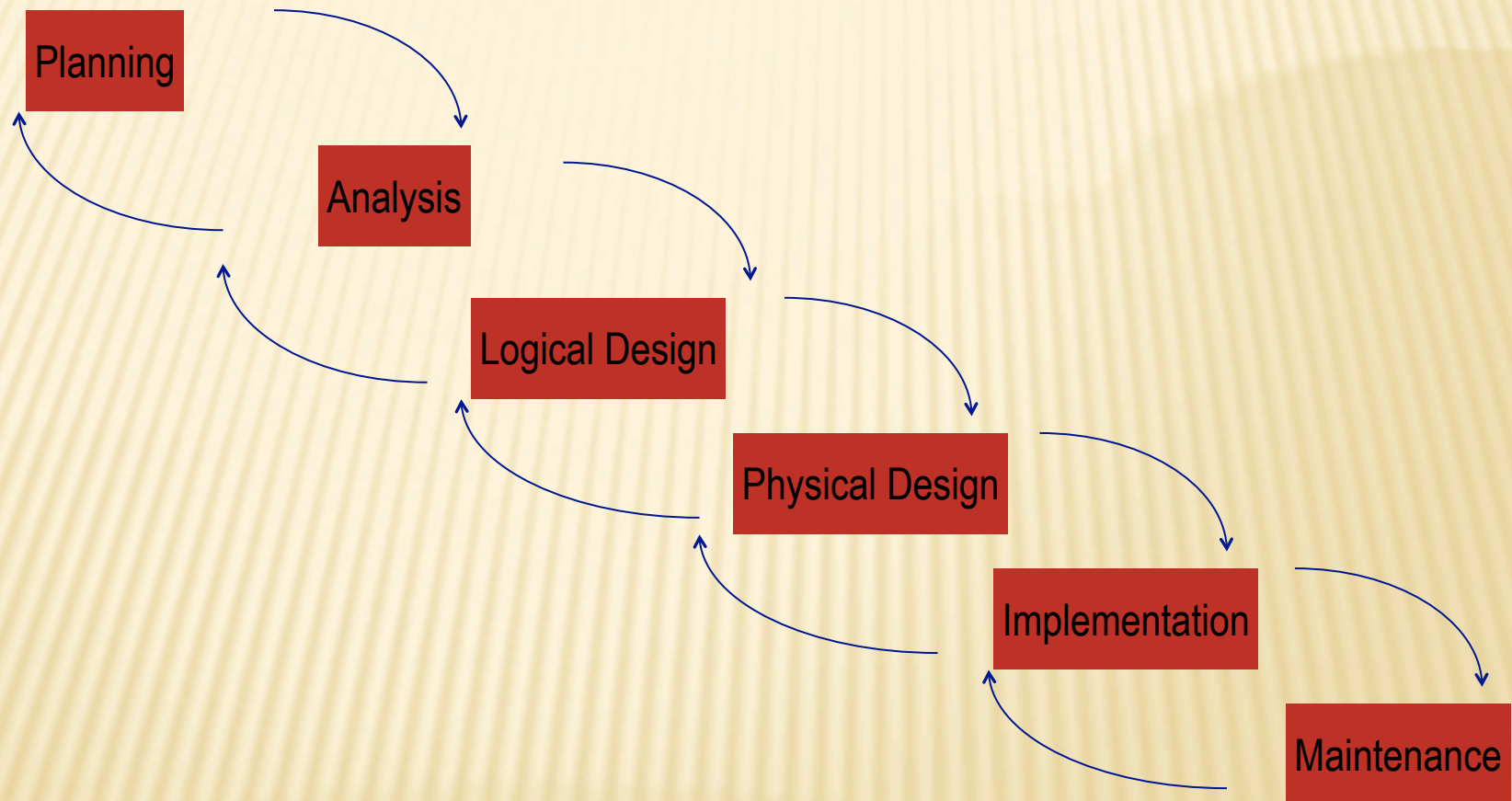
## ✗ SDLC

- + System Development Life Cycle
- + Detailed, well-planned development process
- + Time-consuming, but comprehensive
- + Long development cycle

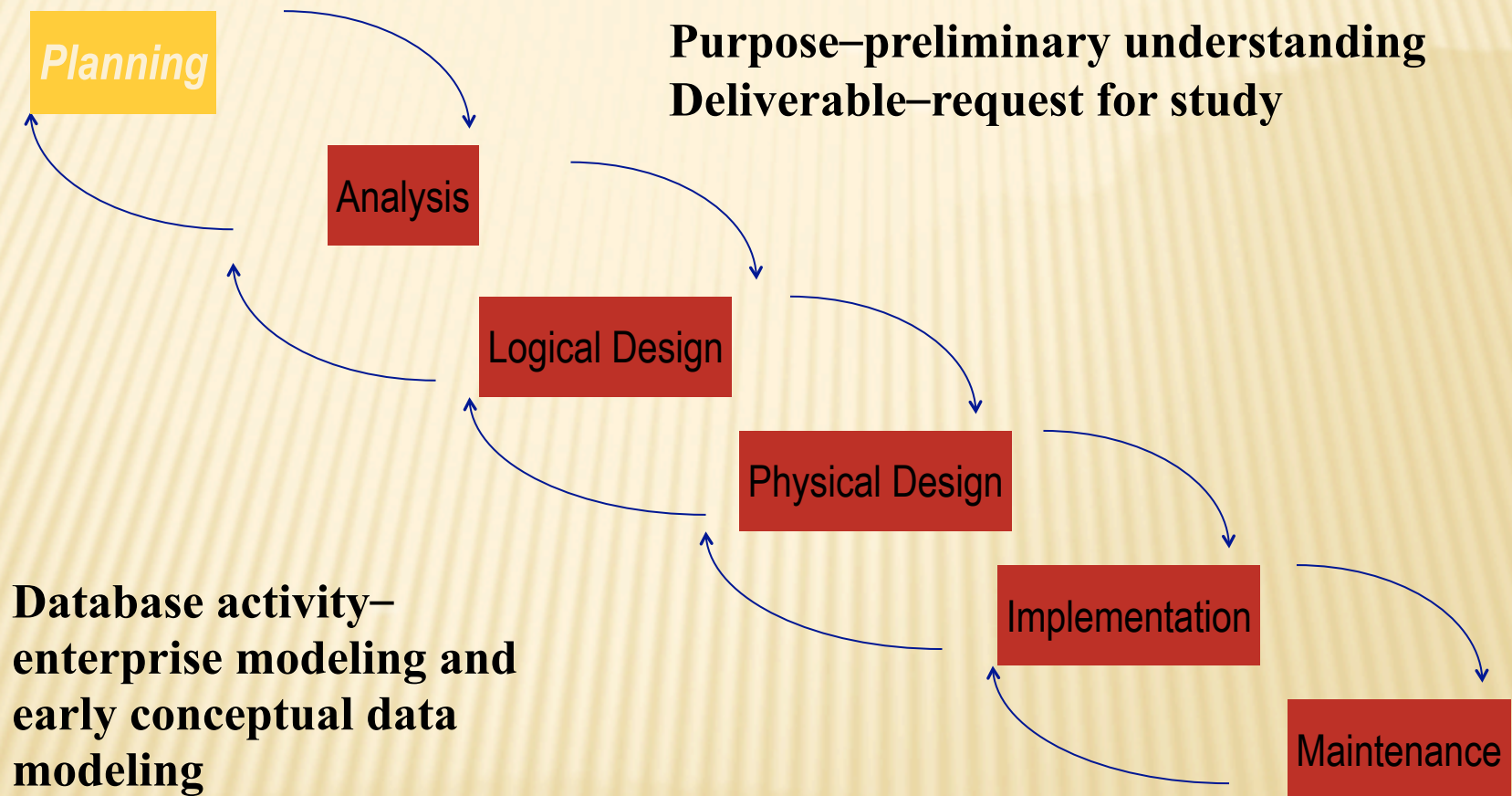
## ✗ Prototyping

- + Rapid application development (RAD)
- + cursory attempt at conceptual data modeling
- + Define database during development of initial prototype
- + Repeat implementation and maintenance activities with new prototype versions

# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1-7)

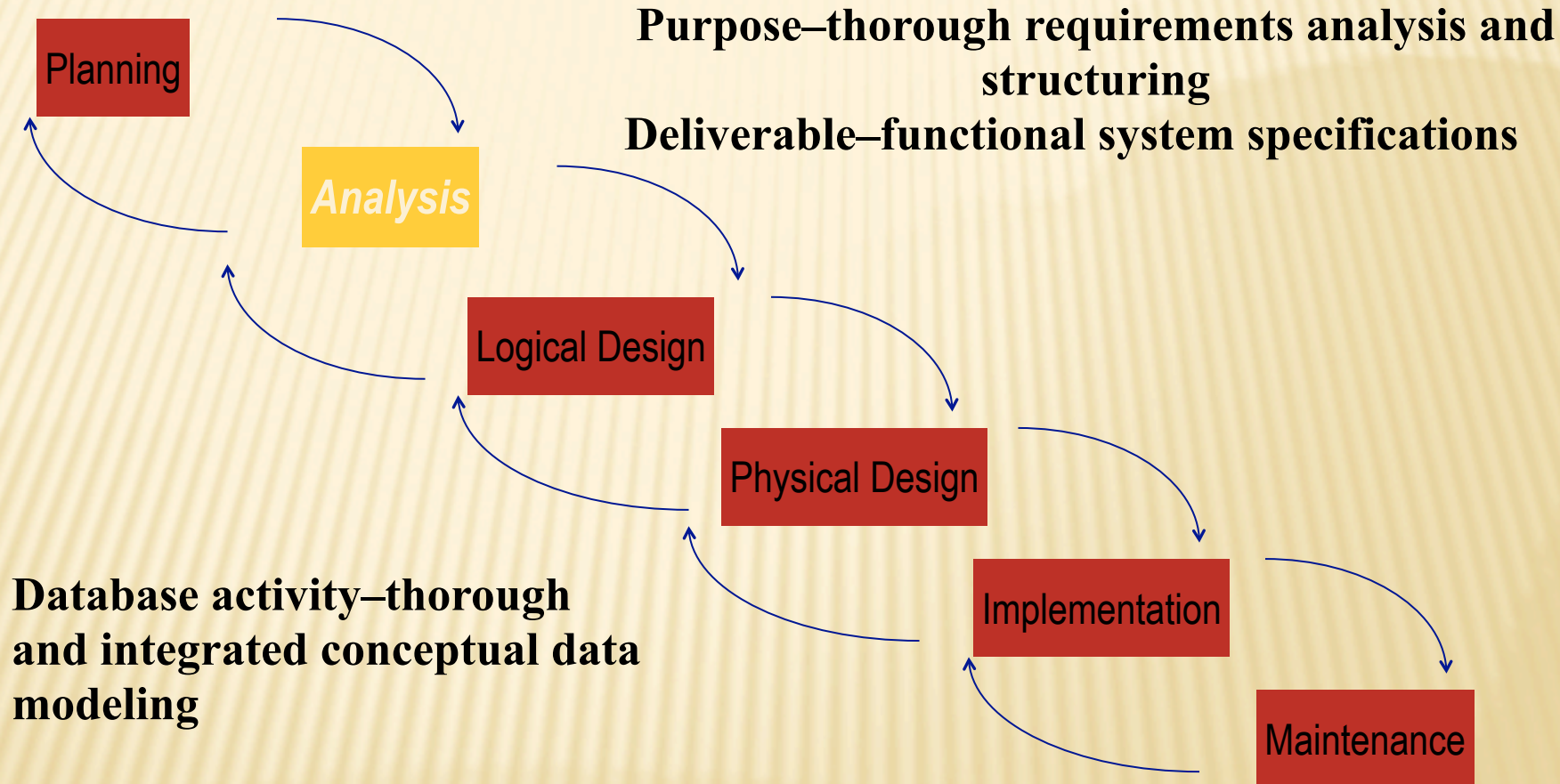


# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1-7) (CONT.)

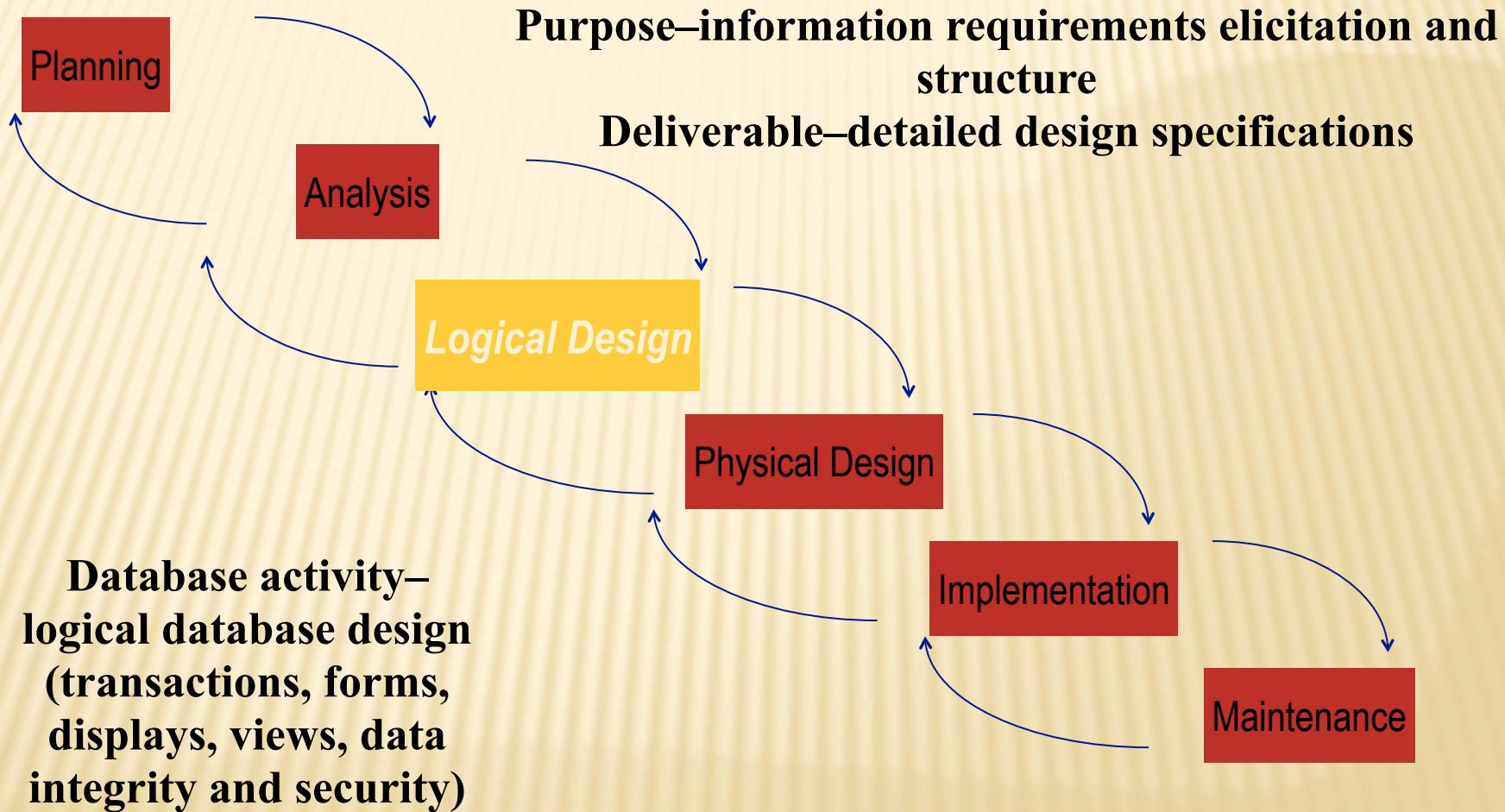




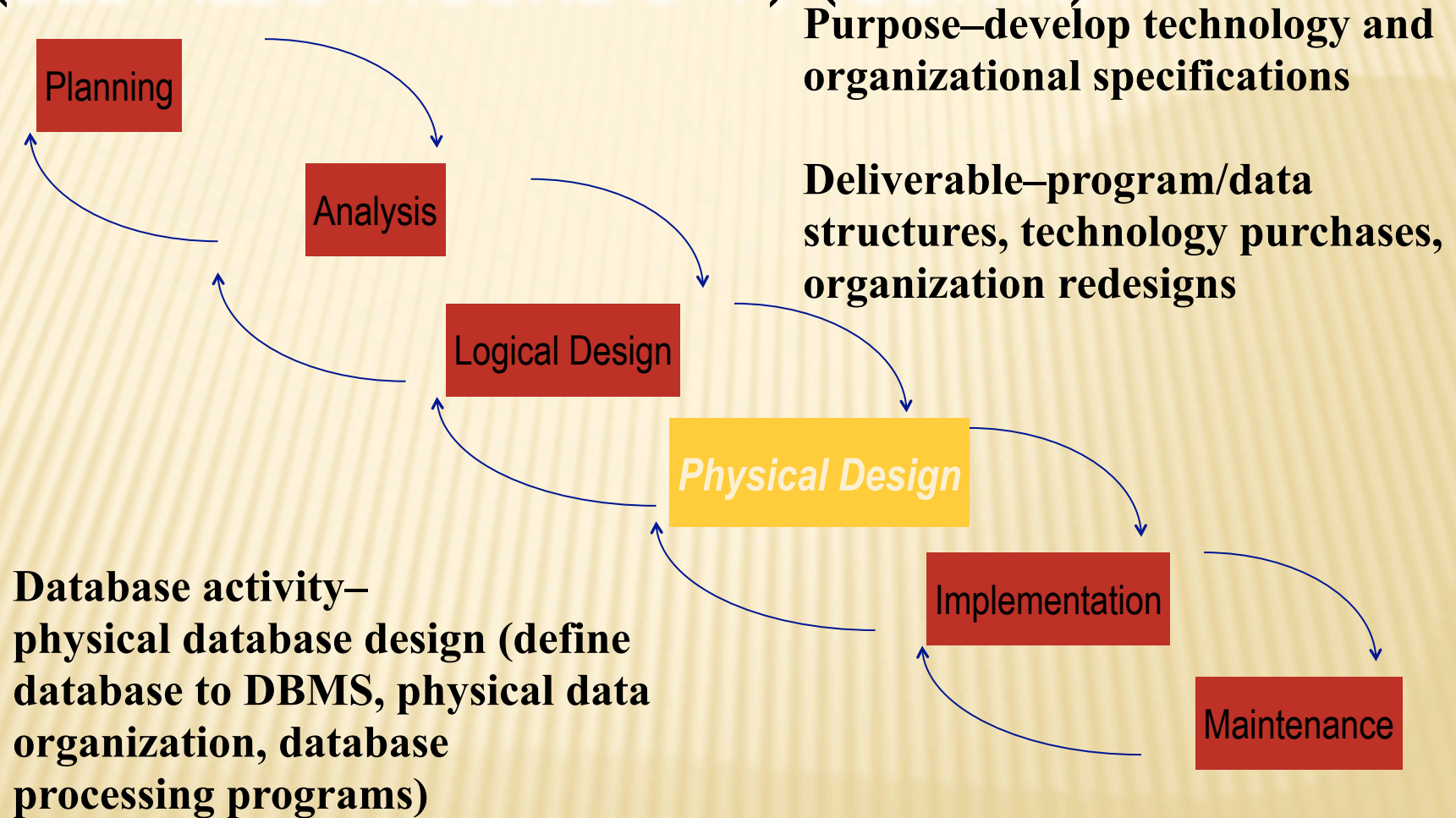
# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1-7) (CONT.)



# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1–7) (CONT.)

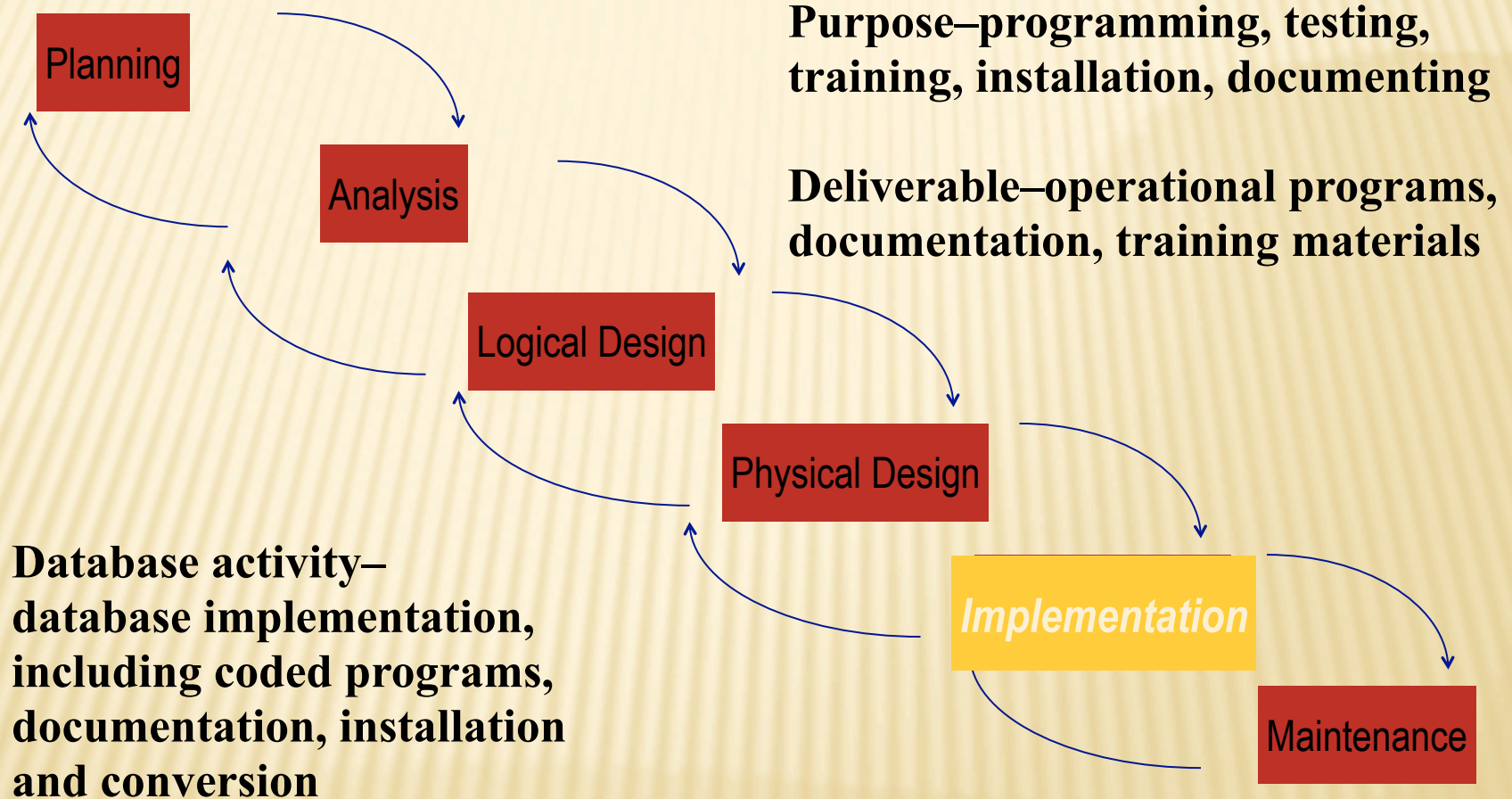


# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1-7) (CONT.)

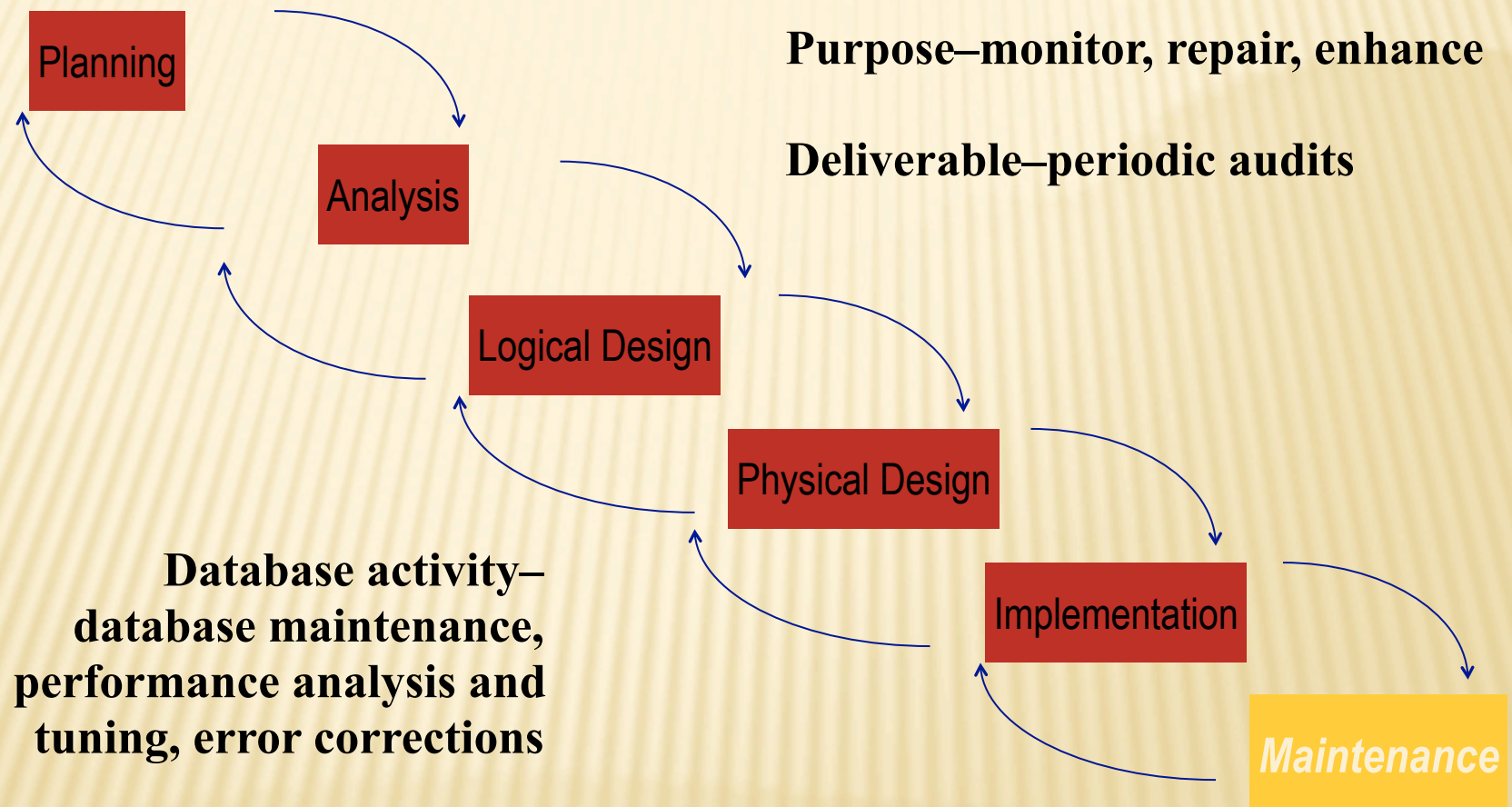




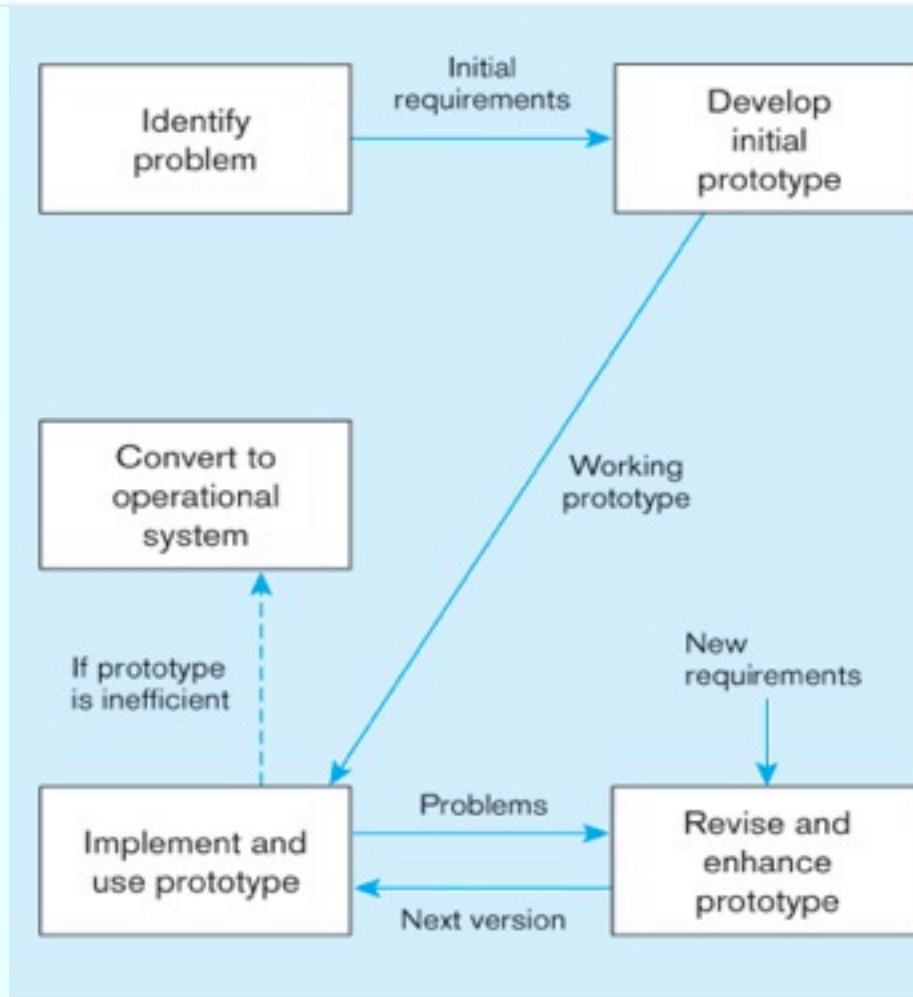
# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1–7) (CONT.)



# SYSTEMS DEVELOPMENT LIFE CYCLE (SEE ALSO FIGURE 1–7) (CONT.)



# Prototyping Database Methodology (Figure 1-8)

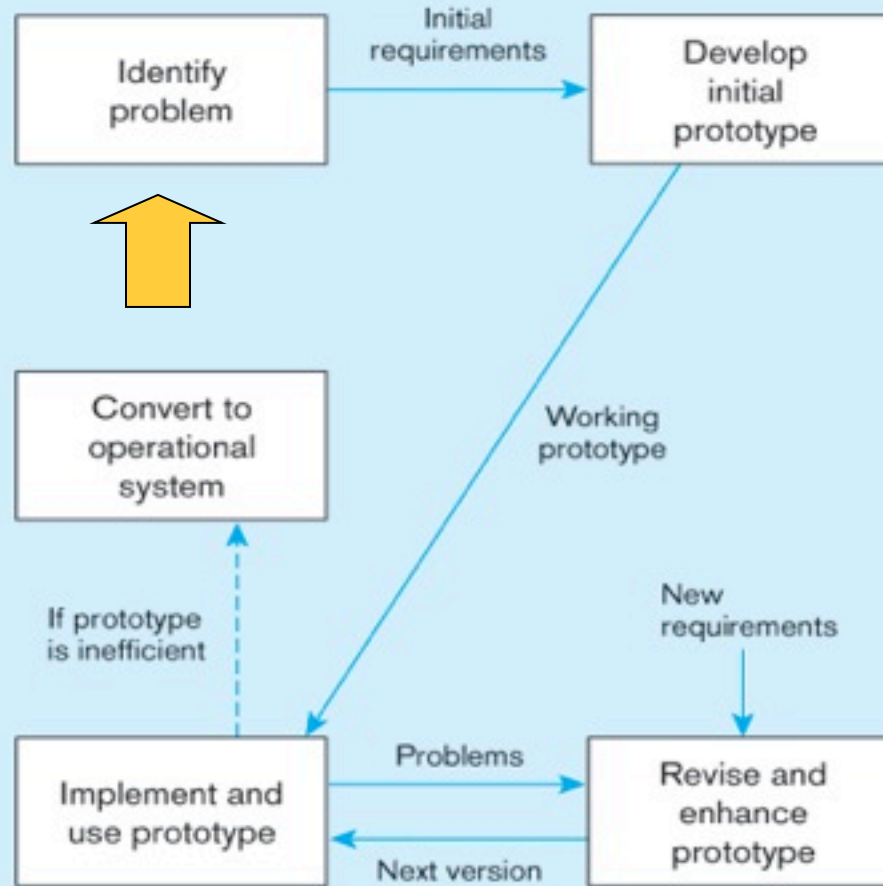




# Prototyping Database Methodology (Figure 1-8) (cont.)

**Conceptual data modeling**

- Analyze requirements
- Develop preliminary data model



# Prototyping Database Methodology (Figure 1-8) (cont.)

## Conceptual data modeling

- Analyze requirements
- Develop preliminary data model

Identify problem

Initial requirements

Develop initial prototype

## Logical database design

- Analyze requirements in detail
- Integrate database views into conceptual data model

## Physical database design and definition

- Define new database contents to DBMS
- Decide on physical organization for new data
- Design database processing programs

## Database implementation

- Code database processing
- Install new database contents, usually from existing data sources

## Database maintenance

- Analyze database to ensure it meets application needs
- Fix errors in database

Convert to operational system

If prototype is inefficient

Implement and use prototype

Problems

Next version

Revise and enhance prototype

New requirements

Working prototype

# Prototyping Database Methodology (Figure 1-8) (cont.)

## Conceptual data modeling

- Analyze requirements
- Develop preliminary data model

Identify problem

Initial requirements

Develop initial prototype

## Logical database design

- Analyze requirements in detail
- Integrate database views into conceptual data model

## Physical database design and definition

- Define new database contents to DBMS
- Decide on physical organization for new data
- Design database processing programs

## Database implementation

- Code database processing
- Install new database contents, usually from existing data sources

## Database maintenance

- Analyze database to ensure it meets application needs
- Fix errors in database

Convert to operational system

If prototype is inefficient

Implement and use prototype

Problems

Revise and enhance prototype

Next version

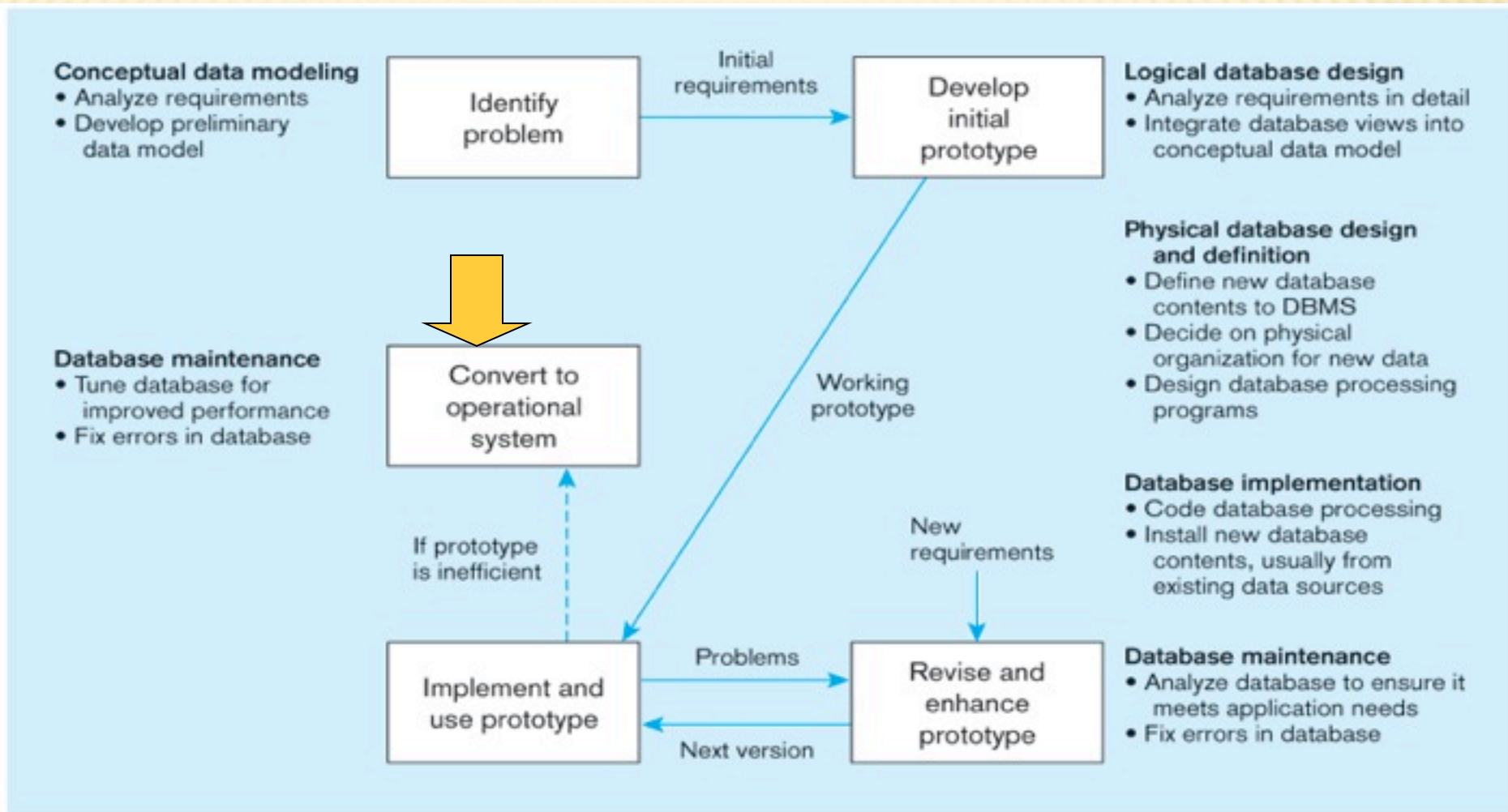
Working prototype

New requirements





# Prototyping Database Methodology (Figure 1-8) (cont.)



# DATABASE SCHEMA

---

## ✖ External Schema

- + User Views
- + Subsets of Conceptual Schema
- + Can be determined from business–function/  
data entity matrices
- + DBA determines schema for different users

## ✖ Conceptual Schema

- + E–R models–covered in Chapters 2 and 3

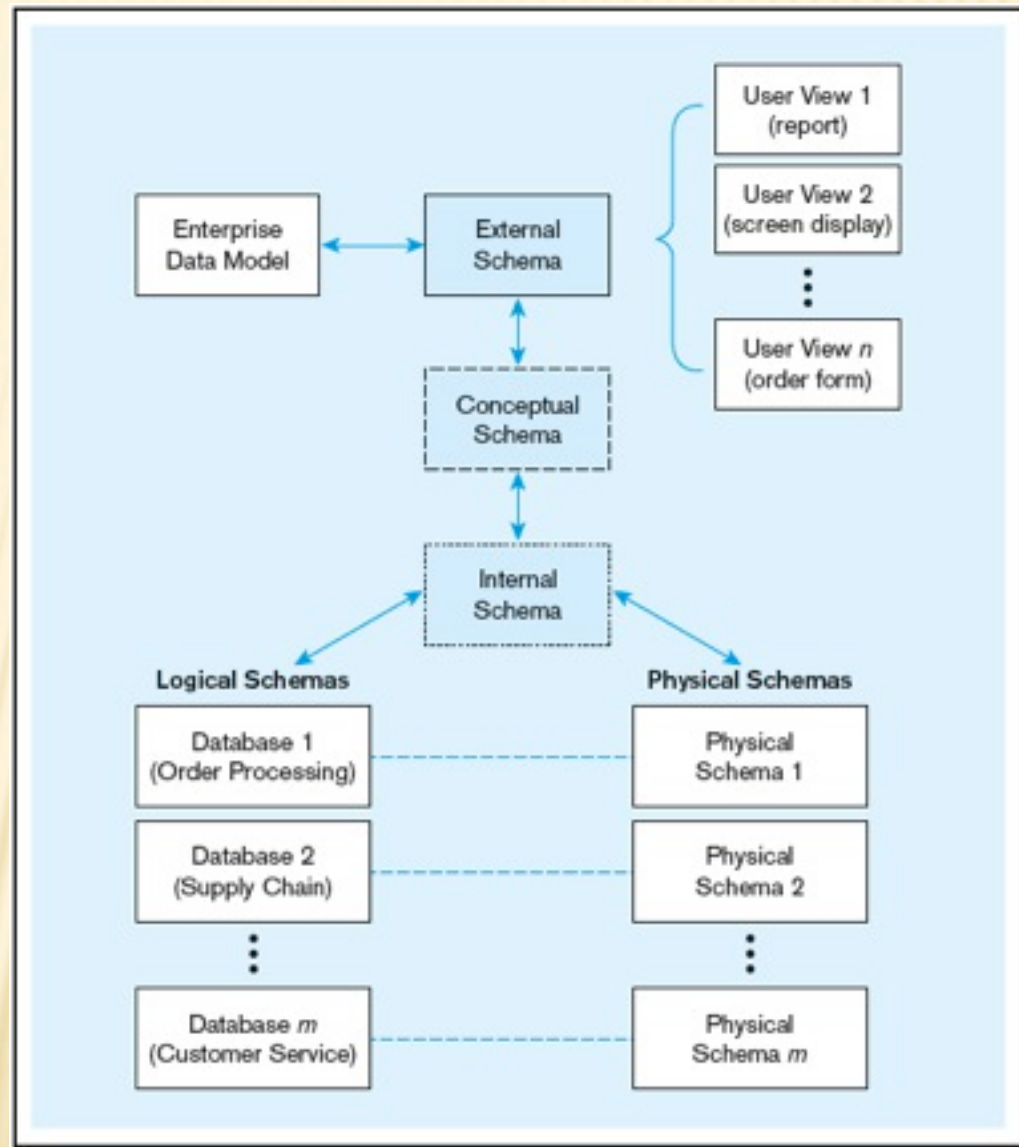
## ✖ Internal Schema

- + Logical structures–covered in Chapter 4
- + Physical structures–covered in Chapter 5

# Figure 1-9 Three-schema architecture

Different people have different views of the database...these are the external schema

The internal schema is the underlying design and implementation





# MANAGING PROJECTS

---

- ✖ Project—a planned undertaking of related activities to reach an objective that has a beginning and an end
- ✖ Initiated and planned in planning stage of SDLC
- ✖ Executed during analysis, design, and implementation
- ✖ Closed at the end of implementation

# MANAGING PROJECTS: PEOPLE INVOLVED

---

- ✗ Business analysts
- ✗ Systems analysts
- ✗ Database analysts and data modelers
- ✗ Users
- ✗ Programmers
- ✗ Database architects
- ✗ Data administrators
- ✗ Project managers
- ✗ Other technical experts

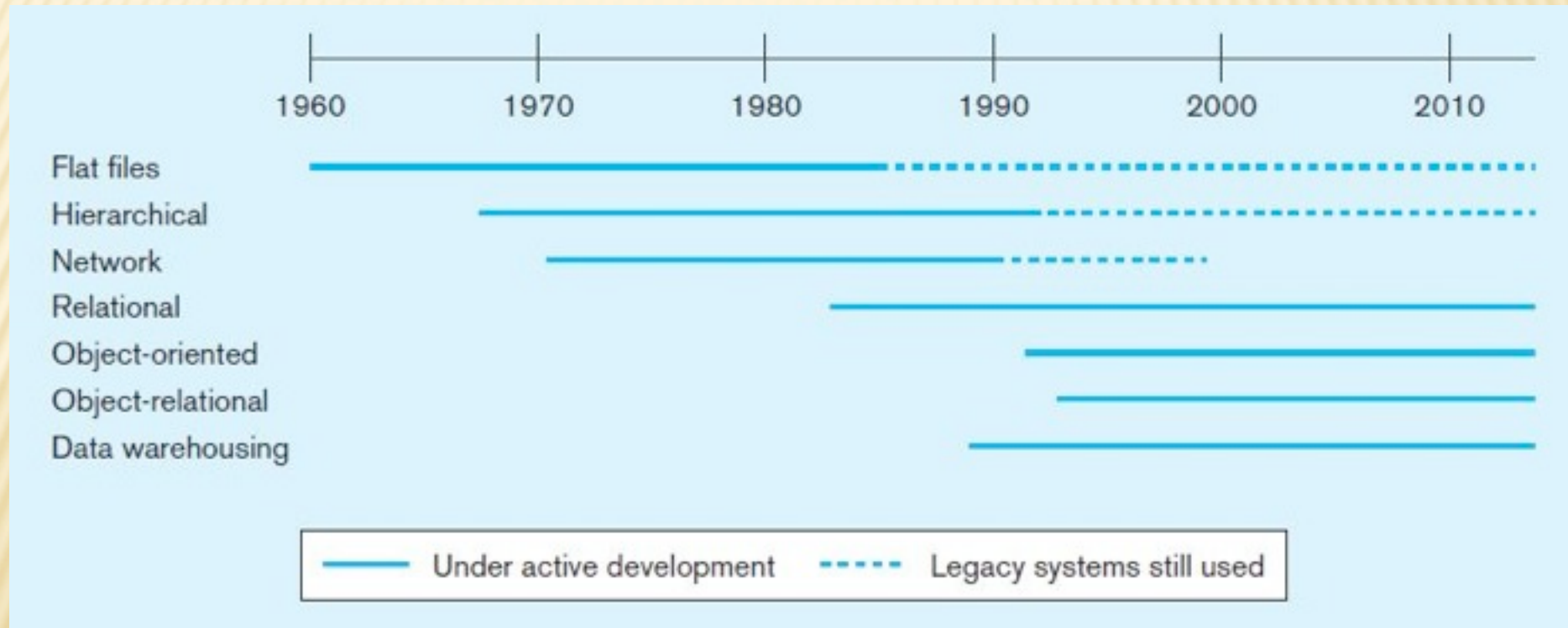
# EVOLUTION OF DATABASE

---

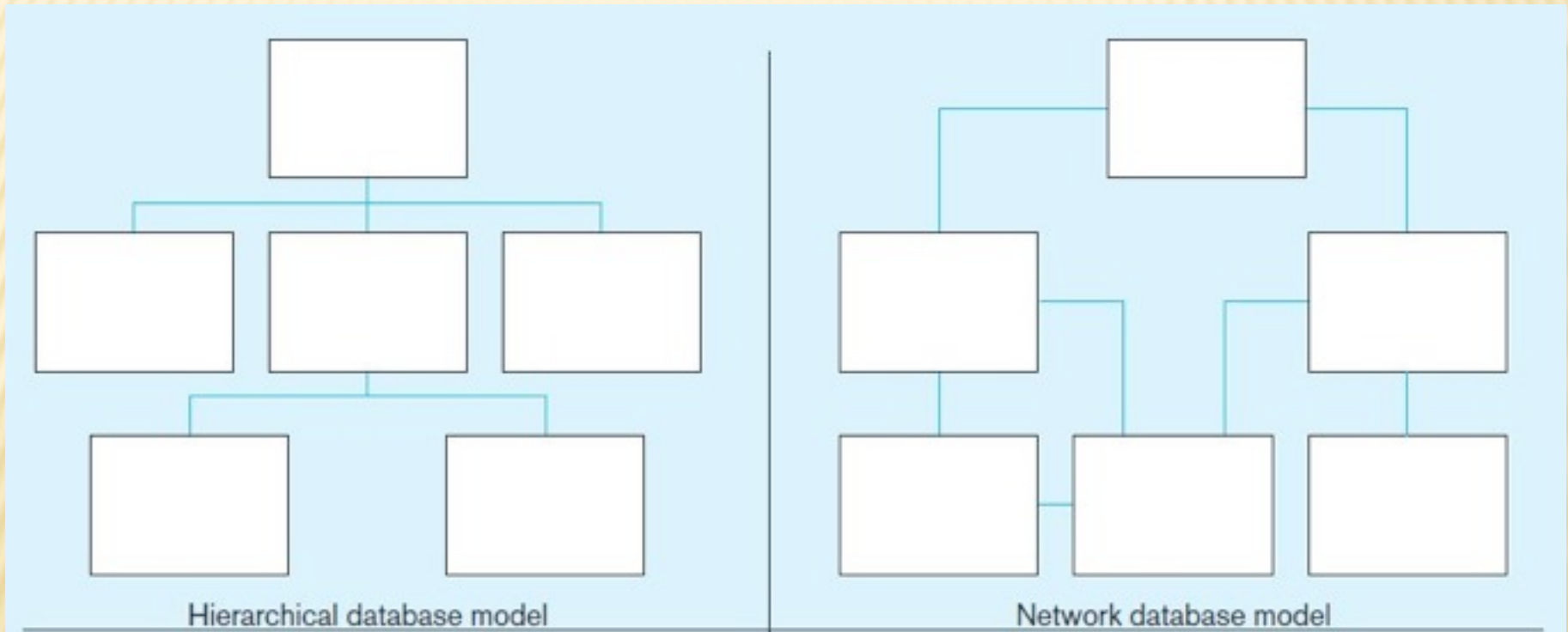
- ✗ Driven by four main objectives:
  - ✗ Need for program–data independence → reduced maintenance
  - ✗ Desire to manage more complex data types and structures
  - ✗ Ease of data access for less technical personnel
  - ✗ Need for more powerful decision support platforms



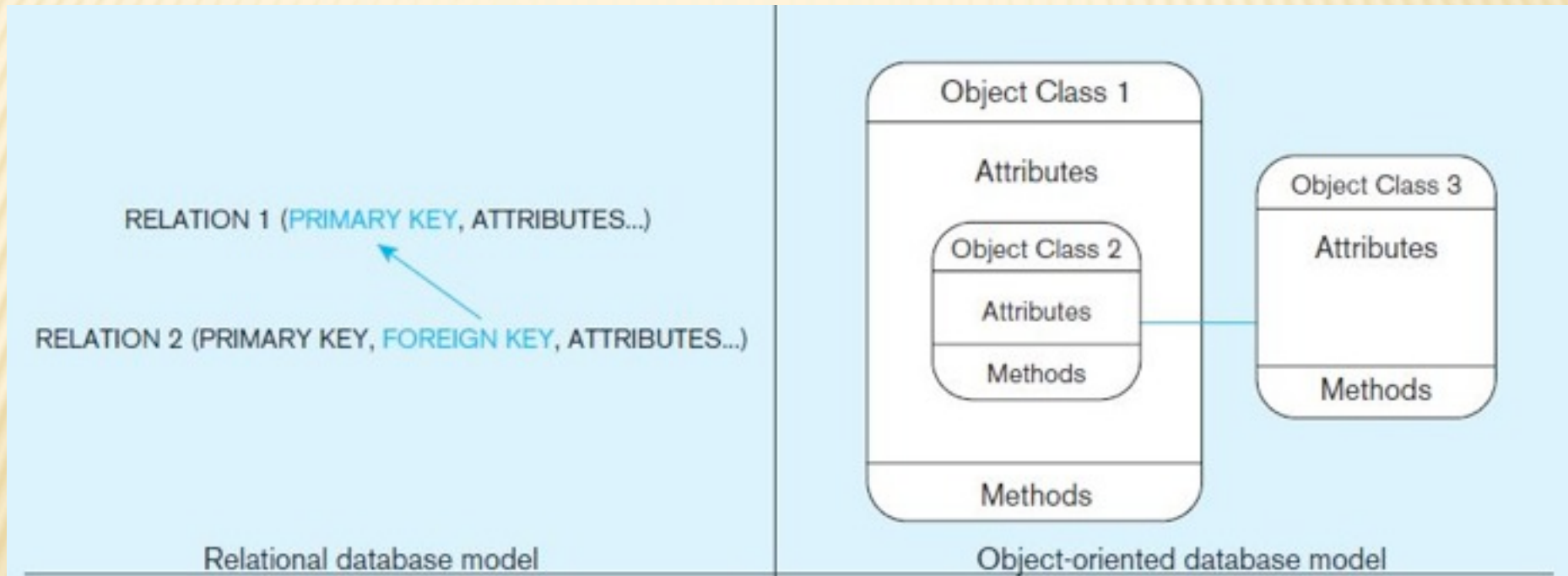
# Figure 1-10a Evolution of database technologies



## Figure 1-10b Database architectures

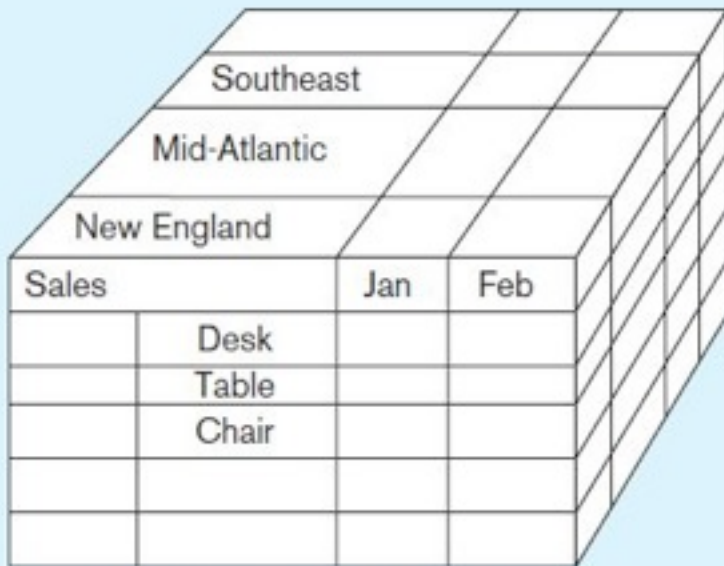


## Figure 1-10b Database architectures (cont.)

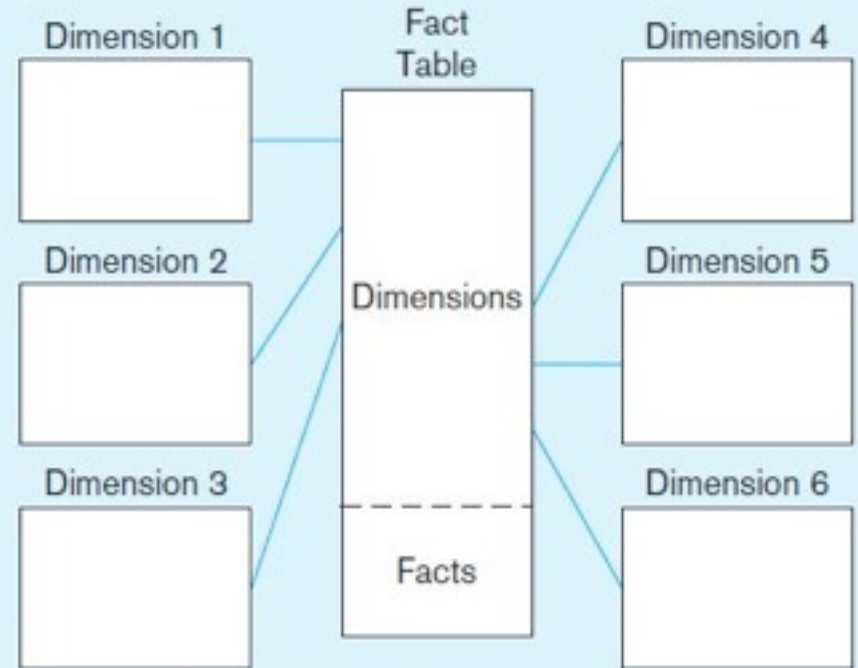




## Figure 1-10b Database architectures (cont.)



Multidimensional database model –  
multidimensional cube view



Multidimensional database model –  
star-schema view

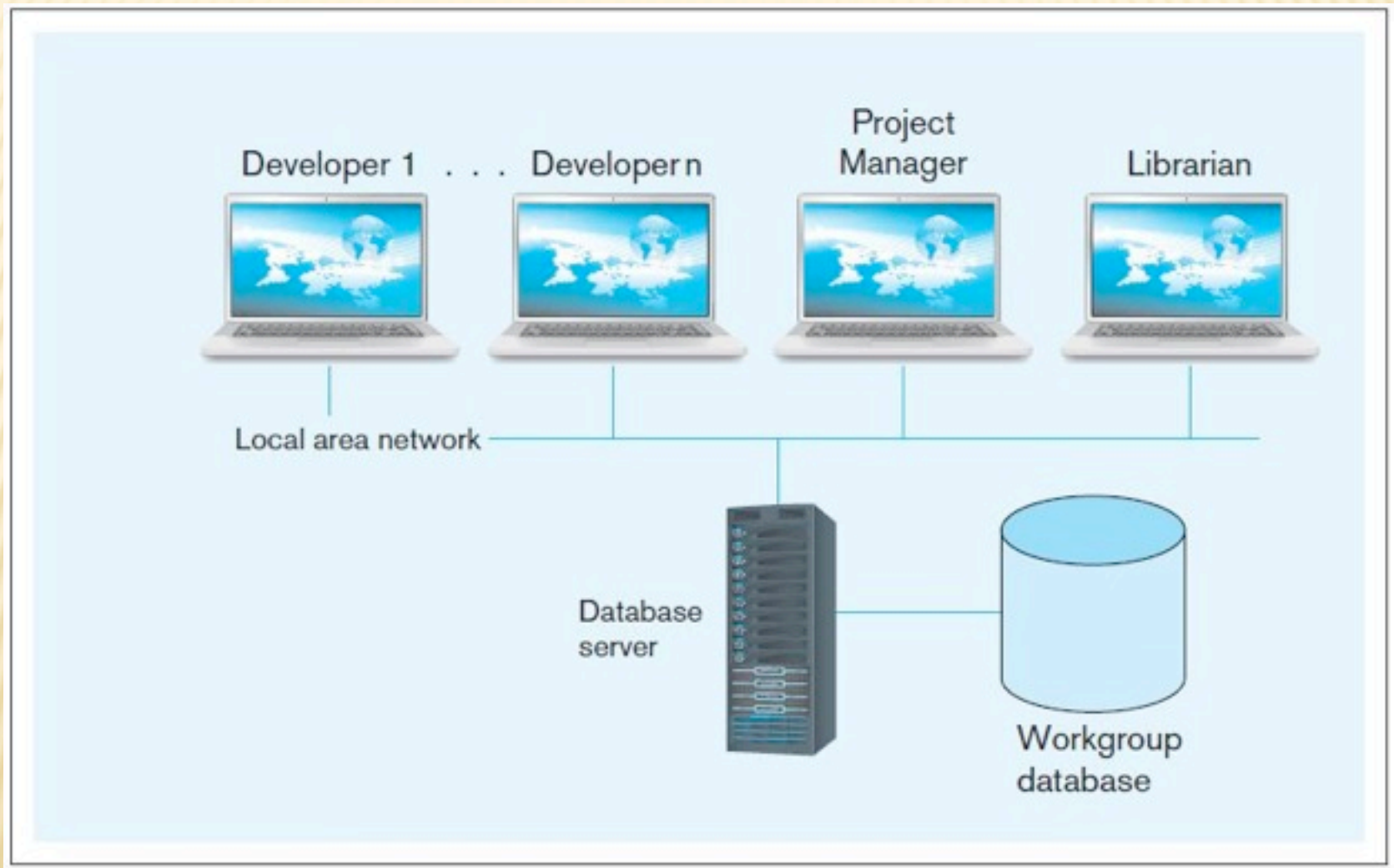
# THE RANGE OF DATABASE APPLICATIONS

- ✖ Personal databases
- ✖ Two-tier and N-tier Client/Server databases
- ✖ Enterprise applications
  - + Enterprise resource planning (ERP) systems
  - + Data warehousing implementations

**TABLE 1-5** Summary of Database Applications

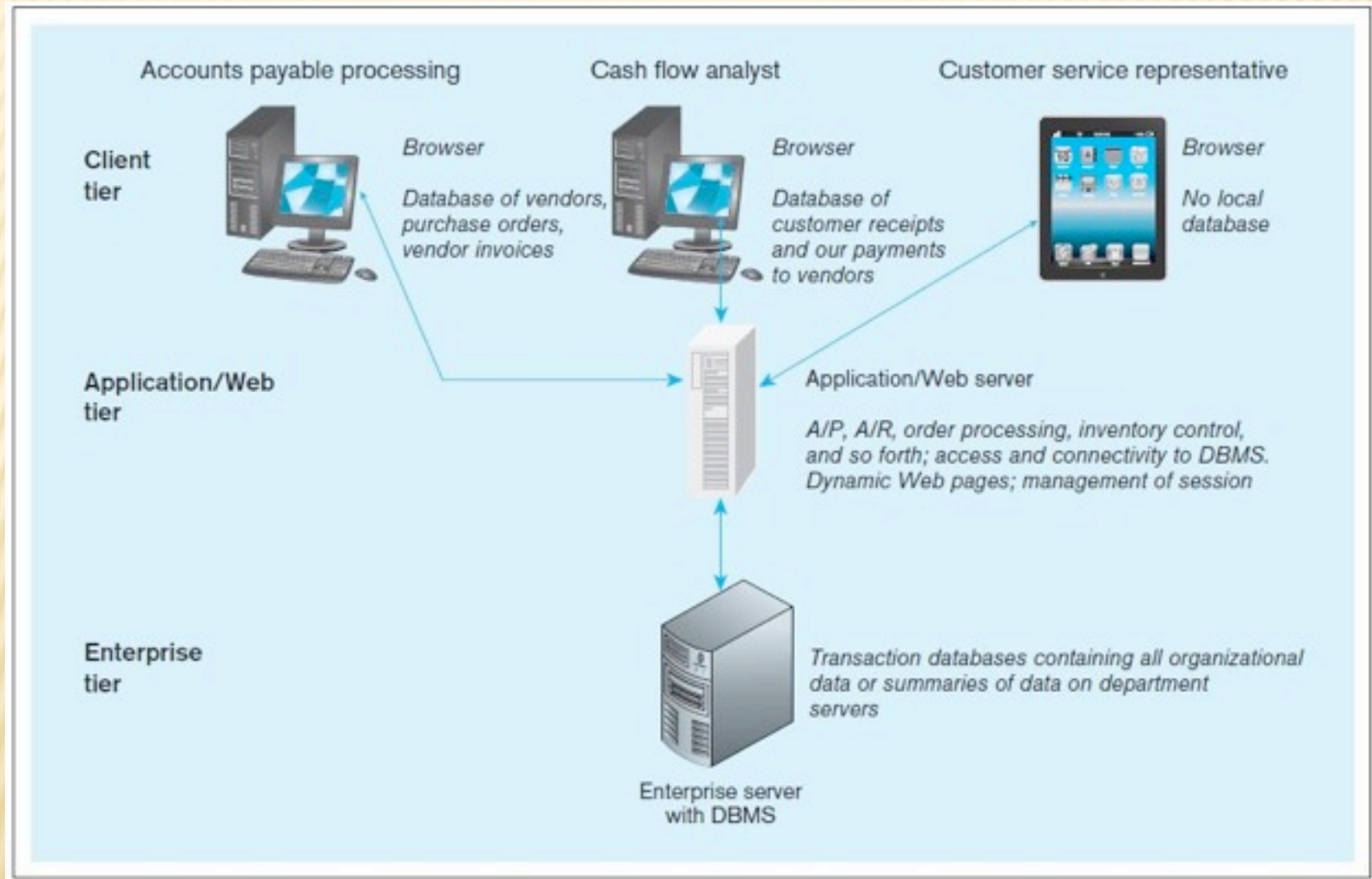
Type of Database / Application	Typical Number of Users	Typical Size of Database
Personal	1	Megabytes
Two-tier	5–100	Megabytes–gigabytes
Three-tier	100–1000	Gigabytes
Enterprise resource planning	>100	Gigabytes–terabytes
Data warehousing	>100	Terabytes–petabytes

# Figure 1-11 Two-tier database with local area network





# Figure 1-12 Three-tiered client/server database architecture



# ENTERPRISE DATABASE APPLICATIONS

---

- ✖ Enterprise Resource Planning (ERP)
  - + Integrate all enterprise functions (manufacturing, finance, sales, marketing, inventory, accounting, human resources)
- ✖ Data Warehouse
  - + Integrated decision support system derived from various operational databases

**FIGURE 1-13 Computer System for Pine Valley Furniture Company**

